Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 75

Konkurs Oddziału Gdańskiego PTETiS na najlepsze prace dyplomowe

Gdańsk 2021/2022

doi: 10.32016/1.75.02

SYMULACJA AUTOMATU KOMÓRKOWEGO „GRA W ŻYCIE" JOHNA CONWAYA NA WYBRANYM MIKROKONTROLERZE

Paweł BURDZIŃSKI[1], Paweł POCZEKAJŁO[2]

1. Politechnika Koszalińska
   e-mail: u15582@s.tu.koszalin.pl
2. Politechnika Koszalińska, Wydział Elektroniki i Informatyki
   tel.: +48 94 3478 721 e-mail: pawel.poczekajlo@tu.koszalin.pl

Streszczenie: Wykonany prototyp pozwala na symulację automatu komórkowego „Gry w Życie" Johna Conwaya oraz innych wariantów dwuwymiarowych automatów bazujących na liczbie żywych komórek w sąsiedztwie Moore'a. Urządzenie umożliwia wczytywanie podstawowych struktur z pamięci mikrokontrolera oraz dowolnych map z ustawieniami reguł symulacji z karty microSD. Struktury można edytować, tworzyć własne lub generować losowo. Sterowanie odbywa się pilotem na podczerwień, a symulacja jest wyświetlana poprzez przejściówkę VGA. Implementacja programu pozwala na symulację dwóch niezależnych automatów komórkowych jednocześnie, przełączanie istnienia ścian na krawędzi mapy oraz inne drugorzędne opcje.

Słowa kluczowe: automat komórkowy, Gra w Życie, John Conway, Arduino Due.

1. WPROWADZENIE

Automaty komórkowe pozwalają na przeprowadzanie złożonych symulacji za pośrednictwem prostych reguł [1-3]. Niektóre z nich posiadają kompletność Turinga, co pozwala na przeprowadzenie dowolnej operacji komputerowej wewnątrz symulacji, w tym zaprojektowanie automatu komórkowego wewnątrz niego samego. Przykładem może być dobrze znany dwuwymiarowy, dwustanowy automat komórkowy „Gra w Życie" Johna Conwaya. Symulacja odbywa się na płaszczyźnie podzielonej na kwadratowe pola. Każde pole jest komórką, która może przyjąć jeden z dwóch stanów „żywy" lub „martwy". Następny stan komórek obliczany jest równolegle w tej samej jednostce czasu. Kolejny stan jest zależny od liczby „żywych" sąsiadów w sąsiedztwie Moore'a – najbliższe 8 pól otaczające komórkę. Reguły dla symulacji „Gry w Życie" można przedstawić w jedynie dwóch punktach:

1. Każda żywa komórka z dwoma lub trzema sąsiadami pozostaje nadal żywa, w przeciwnym razie umiera.
2. Każda martwa komórka z dokładnie trzema sąsiadami – rodzi się.

Pomimo tak prostych zasad, „Gra w Życie" pozwala na symulowanie złożonych struktur oraz operacji za ich pośrednictwem nawet rozpoczynając symulację z losowym wypełnieniem płaszczyzny żywymi komórkami.

1.1. Założenia projektu

Do zrealizowania prototypu wybrano moduł Arduino DUE z 32-bitowym mikrokontrolerem z rodziny ARM pozwalającym na pracę z sygnałem zegarowym o częstotliwości 84 MHz. Wybrana płytka deweloperska umożliwia programowanie mikrokontrolera w intuicyjnym środowisku programistycznym Arduino IDE, które zawiera w sobie wiele pomocnych bibliotek. Jedną z najważniejszych bibliotek zastosowanych w projekcie jest „Due VGA" [4] – biblioteka odpowiedzialna za generowanie sygnału VGA z płytki Arduino DUE oraz używanie 8-bitowych kolorów przy maksymalnej rozdzielczości 320 × 240. Postanowiono, żeby sterowanie interfejsem użytkownika odbywało się za pomocą pilota na podczerwień, a mapy wraz z regułami symulacji będzie można wczytać z karty pamięci microSD.

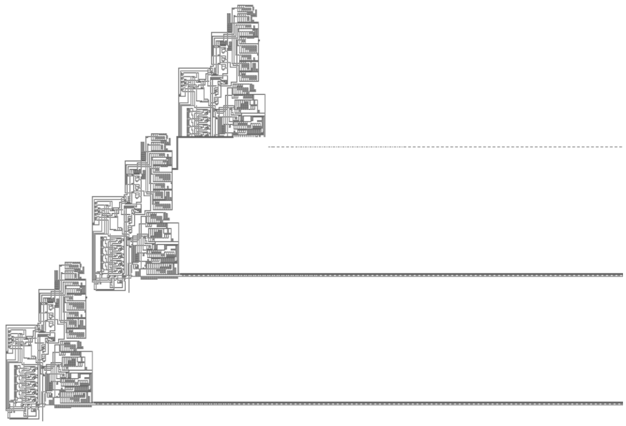
1.2. Automaty komórkowe

Powstanie pierwszych automatów komórkowych można przypisać na lata 50. XX wieku, kiedy polski naukowiec Stanisław Marcin Ulam zajmujący się matematyką oraz fizyką jądrową wspólnie z Johnem von Neumannem – węgierskim matematykiem, fizykiem, inżynierem chemikiem oraz informatykiem – opracowali metodę obliczania ruchu cieczy [5]. Zastosowana metoda polegała na podziale cieczy na grupę komórek, które posiadały skończoną liczbę stanów, a ich następny stan był obliczany na postawie stanów sąsiednich komórek.

John von Neumann zachwycał się ideą samo-replikujących się maszyn. Węgierski matematyk zaprojektował samo-replikującą się maszynę w środowisku automatów komórkowych – „Uniwersalny konstruktor Johna von Neumanna" – bez użycia komputera, a szczegóły opisał w książce „Theory of Self-Reproducing Automata", która została dokończona po śmierci informatyka przez Artura Waltera Burksa [6].

Na rysunku 1 można zaobserwować trzy generacje maszyn, które tworzą własną strukturę nad swoją strukturą. Na samym dole znajduje się najmłodsza struktura, która zakończyła w pełni replikację powyżej. Struktura pośrodku nadal pracuje nad generacją samej siebie powyżej. Trzy linie odbiegające od maszyn na prawo są zakodowaną sekwencją instrukcji – tak jak zbiór stanów na taśmach w maszynie Turinga.

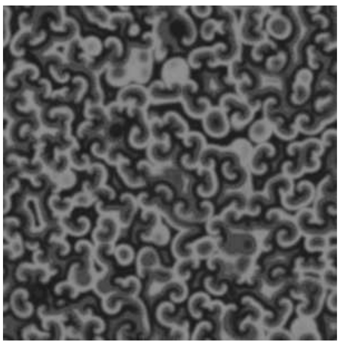
Poza zastosowaniem automatów komórkowych w fizyce można znaleźć ich wykorzystanie również w biologii (rys. 2.), chemii (rys. 3.), kryptografii, generowaniu liczb pseudolosowych [7] czy generowaniu grafiki oraz muzyki [8] jak i w wielu innych dziedzinach.



Rys. 1. Pierwsza implementacja samo-replikującego się uniwersalnego konstruktora Von Neumanna (w 32 stanowym środowisku)



Rys. 2. Muszla stożka tekstylnego – jednego z najbardziej jadowitych ślimaków świata – ukazuje wzór podobny do automatu komórkowego „Rule 30” [10]



Rys. 3. Oscylacyjna reakcja chemiczna Biełousowa-Żabotyńskiego może zostać zasymulowana komputerowo za pomocą automatu komórkowego

Zastosowanie w kryptografii może być przydane przy uzyskiwaniu klucza publicznego – korzystając z automatów komórkowych jako funkcji jednostronnej, których procesów nie można odwrócić lub są bardzo trudne do odwrócenia. Automaty komórkowe znalazły swoje zastosowanie również w pamięciach ECC (Error Correction Code) [9] od których oczekuje się niezawodnego działania i większego bezpieczeństwa danych. Pamięci ECC stosuje się na przykład w miejscach, gdzie jest zwiększony poziom promieniowania jonizującego, które mogłoby zmienić wartość poszczególnych bitów.

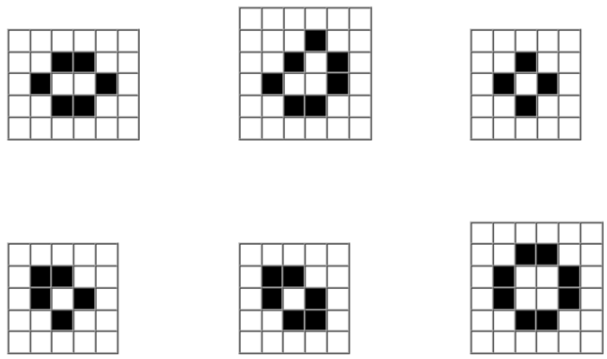
Automaty komórkowe zarówno jak automaty skończone, zawierają w sobie wszystkie predefiniowane reguły dla poszczególnych stanów systemu, a tę skończoną ilość stanów i reguł wykorzystują dla poszczególnych komórek znajdujących się w przestrzeni o skończonej liczbie wymiarów. Każda komórka posiada skończoną liczbę stanów, w jakich może występować. Reguły opisują, jak stan komórek zmienia się w kolejnej dyskretnej jednostce czasu.

1.3. Gra w Życie

Popularność automatów komórkowych znacząco wzrosła w 1970 roku, kiedy angielski matematyk John Conway zaproponował bardzo proste reguły, które pozwalały na generowanie bardzo złożonych struktur. Profesor matematyki nazwał swój automat komórkowy „Grą w Życie” rozgrywający się na dwu wymiarowej przestrzeni oraz posiadający dwa stany komórek: żywa oraz martwa. „Gra w Życie” miała swój własny artykuł w magazynie „Scientific American” napisany przez Martina Gardnera [11] – amerykańskiego pisarza popularnomatematycznych oraz popularnonaukowych artykułów.

Reguły automatu komórkowego Johna Conwaya można uprościć do zapisu „23/3”. Przed ukośnikiem podano wymagane liczby żywych komórek w sąsiedztwie Moore’a, dla których żywe komórki pozostaną nadal żywe w następnej jednostce czasu – dla „Gry w Życie” są to odpowiednio wartości 2 oraz 3. Po ukośniku – podawane są liczby żywych komórek, dla których martwe komórki stają się żywe (wartość 3 dla omawianego automatu).

Struktury ze względu na ich zachowanie można podzielić między innymi na: niezmienne, oscylatory, statki, działa, niestałe oraz Ogrody Edenu. Jedną z najprostszej kategorii struktur mogą być struktury niezmienne – nie ulegają żadnym zmianom w kolejnych jednostkach czasu – a każda komórka podtrzymuje życie sąsiednich. Oscylatory zaś to struktury zmieniające swoją formę w czasie, ale wracające do stanu początkowego po określonej stałej liczbie generacji. Statkami nazywamy oscylatory, które po każdym cyklu zmieniają swoją pozycję, wędrując w jednym kierunku.



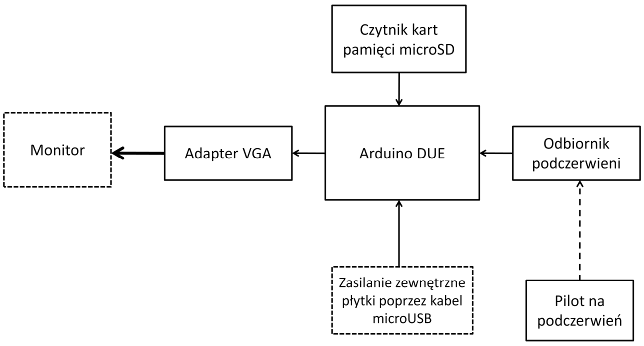
Rys. 4. Kilka przykładów struktur niezmiennych, w górnym rzędzie w kolejności od lewej do prawej: „Hive” (ul), „Loaf” (bochenek), „Tub” (koniczynka), a w dolnym rzędzie: „Boat” (łódź), „Ship” (statek) oraz „Pond” (staw)

2. BUDOWA UKŁADU

Do realizacji projektu zostały wykorzystane następujące moduły:

1. Płytka Arduino DUE z mikrokontrolerem z rodziny ARM,
2. Adapter VGA,

3. Czytnik kart pamięci microSD,
4. Odbiornik podczerwieni,
5. Pilot na podczerwień.



Rys. 5. Schemat opracowanego systemu

2.1. Odbiornik podczerwieni

Moduł odbiornika podczerwieni CHQ1838 działający na częstotliwości 38 kHz potrzebuje zasilania od 2,7 V do 5,5 V – Arduino DUE działające na logice 3,3 V poradzi sobie z zasilaniem oraz odbiorem sygnału z modułu. Wybrany moduł posiada również diodę LED, która będzie sygnalizowała moment wykrycia podczerwieni. Wyjście modułu można podłączyć do dowolnego wejścia cyfrowego mikrokontrolera (w tym przypadku użyto pinu 7).

Tabela 1. Podłączenie modułu odbiornika podczerwieni do mikrokontrolera oraz odpowiadające im piny

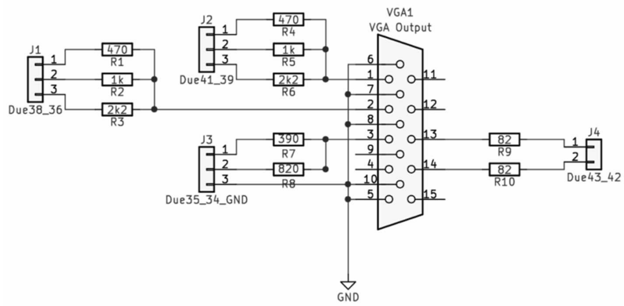
Arduino DUE	Moduł odbiornika
GND	GND
3,3 V	VCC
D7	S

2.2. Czytnik kart pamięci microSD

Zastosowany moduł czytnika kart pamięci microSD zapewnia komunikację poprzez interfejs SPI oraz pracuje z napięciem od 3,3 V do 5 V i wymaga zasilania od 4,5 V do 5,5 V. Płytka Arduino DUE posiada pin 5 V, który poprawnie zasili moduł. Komunikacja z modułem będzie odbywała się na logice 3,3 V.

Tabela 2. Podłączenie modułu czytnika kart pamięci microSD do mikrokontrolera oraz odpowiadające im piny

Arduino DUE	Moduł czytnika kart
GND	GND
5 V	VCC
MISO (D74)	MISO
MOSI (D75)	MOSI
SCK (D76)	SCK
SD-CS (D4)	CS



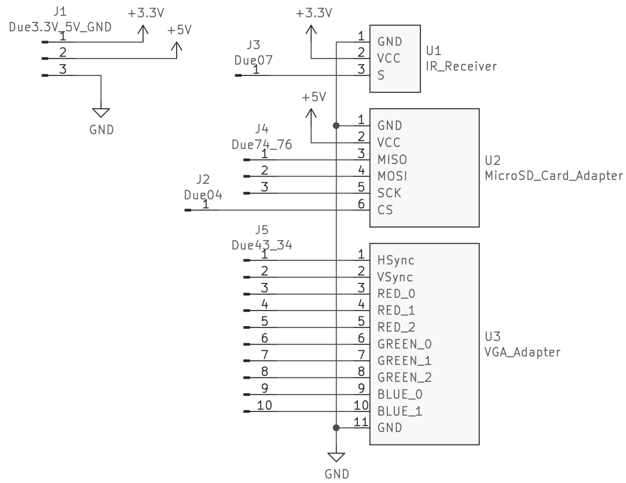
Rys. 6. Schemat adaptera oraz odpowiadające im piny mikrokontrolera [12]

2.3. Adapter VGA

Zastosowany adapter VGA przekształca sygnał cyfrowy z Arduino DUE na sygnał analogowy.

2.4. Schemat połączeń

Wszystkie moduły oraz ich odpowiednie połączenia z płytką deweloperską Arduino DUE zostały zaprezentowane na rysunku 7.



Rys. 7. Schemat elektroniczny zbudowanego układu

3. ZAPROGRAMOWANIE AUTOMATU KOMÓRKOWEGO NA MIKROKONTROLERZE

3.1. Wstępne przygotowania

Przed rozpoczęciem programowania symulacji automatu komórkowego należało zainstalować obsługę płytki i odpowiednie biblioteki w Arduino IDE, odczytać kody dla poszczególnych przycisków pilota oraz ustalić dokładne założenia dotyczące interfejsu użytkownika przed rozpoczęciem implementacji.

Użyte biblioteki w projekcie:

1. <VGA.h> Due VGA napisana przez użytkownika GitHub'a „stimmer"
2. <IRremote2.h> napisana przez użytkownika GitHub'a „enternoescape"
3. <SPI.h> wbudowana w Arduino IDE
4. <SD.h> wbudowana w Arduino IDE

Założono, że po podłączeniu urządzenia do zasilania na monitorze wyświetli się menu z wyborem następujących opcji: symulacji losowo wygenerowanej mapy lub jednej z 5 przygotowanych wcześniej map, własne ustawienia symulacji z możliwością podziału ekranu na dwie osobne symulacje z wybranymi regułami lub wczytania mapy z karty pamięci oraz wyświetlenie zdjęcia Johna Conwaya. Postanowiono, że interfejs symulacji będzie składał się z mapy oraz paska z informacjami takimi jak: nazwa mapy bądź struktury, przerwa w milisekundach pomiędzy jednostkami czasowymi symulacji, numer generacji, liczba żywych komórek na mapie. Ustalono takie funkcjonalności jak zatrzymywanie i wznawianie symulacji, modyfikację stanu komórek podczas zatrzymania, ręczne przeskakiwanie do następnego stanu, regulowanie odstępów czasowych pomiędzy kolejnymi stanami, przełączanie wyświetlania siatki, przełączanie istnienia ścian.

3.2. Biblioteka Due VGA

Biblioteka Due VGA pozwala na generowanie sygnału VGA z płytki Arduino DUE oraz obsługuje rozdzielczość

ekranu 320 × 240 w 8-bitowych kolorach. Pierwsze trzy najbardziej znaczące bity przypadają na kolor czerwony, kolejne trzy – zielony, a ostatnie dwa najmłodsze bity odpowiadają kolorowi niebieskiemu. Biblioteka posiada podstawowe funkcje do rysowania oraz wyświetlania tekstu na ekranie.

3.3. Implementacja „Gry w Życie" na mikrokontrolerze

W celu zmieszczenia na ekranie dodatkowych informacji statystycznych o strukturze podczas symulacji postanowiono zostawić na dole ekranu 30 pikseli odstępu na pasek informacyjny. Pozostałe miejsce zostało wykorzystane na wyświetlenie siatki komórek o wymiarze 32 kolumn oraz 21 wierszy w rozmiarze 10×10 pikseli. Dla reguł gry została utworzona struktura danych składająca się z dwóch dziewięcioelementowych tablic wartości typu „bool". Jedna tablica odpowiada liście reguł dla jakich ilości żywych komórek w sąsiedztwie Moore'a, żywe komórki pozostają nadal żywe w następnej jednostce czasu, a druga tablica odpowiada za warunki, kiedy martwa komórka staje się żywa. W ten sposób utworzona struktura tworzy dwie zmienne reguł – dla potrzeby 2 różnych symulacji obok siebie o różnych zasadach. Pozostałe zmienne biorące udział w symulacji wpływają na nią bezpośrednio lub ustawiają wizualny efekt siatki oraz zbierają dane statystyczne.

Symulacja automatu komórkowego w programie odbywa się następująco:

1. Program przechodzi przez każdą komórkę bieżącej generacji oraz uwzględniając opcję ścian i żywotność komórki przypisuje nowy stan dla przyszłej generacji ze zbioru zasad. W przypadku dwóch niezależnych symulacji — uwzględniana jest wartość x (numer kolumny w pętli) dzieląca siatkę na dwie części od wartości 16.
2. Po obliczeniu nowej generacji przepisuje nowe wartości do aktualnej mapy.
3. Zwiększany jest licznik generacji o 1.

3.4. Główna pętla programu

Po wykonaniu metody „setup" program wywołuje metodę „loop", która jest główną pętlą programu. W pętli tej znajdują się 3 główne warunki:

1. Jeżeli odebrano nową wartość z pilota podczerwieni – obsłuż interakcję i zareaguj odpowiednio.
2. Jeżeli symulacja aktualnie się odbywa – sprawdź, czy minęła jednostka czasowa pomiędzy generacjami i zaktualizuj siatkę oraz panel informacyjny w takim przypadku.
3. Jeżeli uruchomiony jest kreator własnych symulacji – sprawdź, czy minęło 100 milisekund, jeżeli tak – zmień kolor podświetlanej opcji dla efektu wizualnego.

Wewnątrz pierwszego warunku instrukcją „switch" uwzględniany jest poszczególny stan programu. Jeżeli wyświetlane jest menu, to użytkownik ma możliwość uruchomienia losowej symulacji lub jednej z domyślnych w programie, przejścia do kreatora własnej symulacji, która pozwala również na wczytywanie map z karty microSD oraz wybrania ostatniej opcji – wyświetlenia zdjęcia Johna Conwaya. W przypadku trwania symulacji, za pomocą pilota można: wznowić lub zatrzymać symulację, zmienić odstęp czasowy pomiędzy generacjami, ręcznie przeskakiwać do następnej generacji, modyfikować stan komórek na mapie. Kiedy kreator symulacji jest uruchomiony, pilot pozwala na wybranie trybu pojedynczego automatu komórkowego lub dwóch niezależnych od siebie, ustalenia zasad symulacji, wczytania mapy z karty pamięci wraz z jej domyślnymi ustawieniami oraz ich ewentualną zmianą, uruchomienia symulacji zgodnie z wcześniej wprowadzonymi opcjami. W przypadku wyświetlania zdjęcia Johna Conwaya możliwy jest tylko powrót do menu – funkcja ta jest możliwa z każdego stanu programu.

4. PODSUMOWANIE

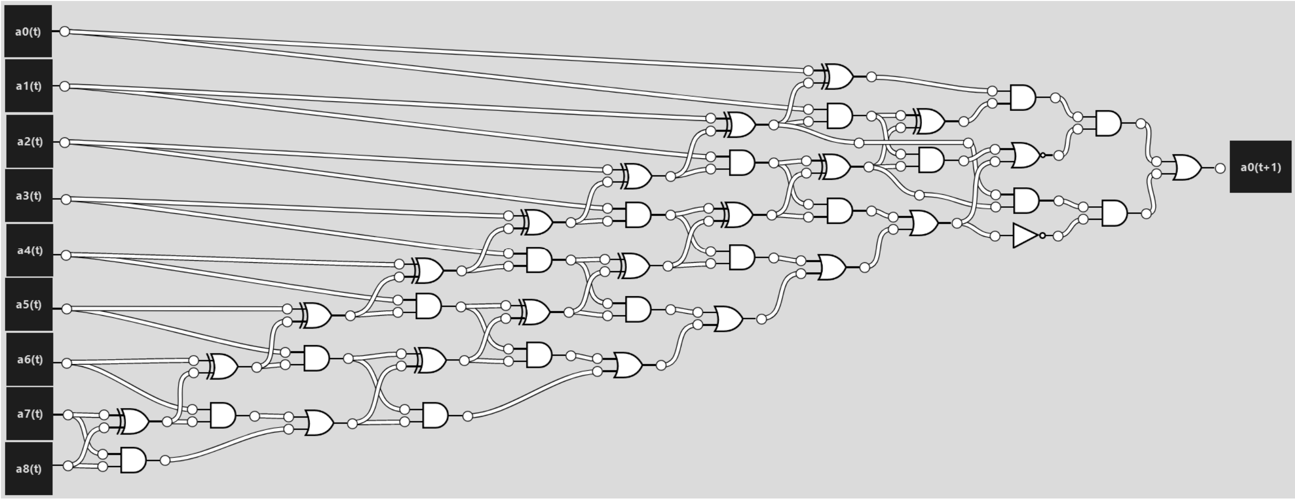
Celem pracy było zrealizowanie prototypu urządzenia z wykorzystaniem wybranego mikrokontrolera do symulacji automatu komórkowego i jego wizualnej reprezentacji na monitorze poprzez złącze VGA.

Automaty komórkowe pokazują, jak proste reguły mogą generować złożone i skomplikowane struktury – tak samo, jak proste bramki logiczne pozwalają na zbudowanie procesorów czy układów FPGA, które potrafią przetworzyć skomplikowane algorytmy z dużą liczbą danych w krótkim czasie. Wykorzystanie automatów komórkowych sięga takich dziedzin jak informatyka, kryptografia, fizyka, biologia czy muzyka. Kompletność Turinga niektórych automatów komórkowych takich jak „Gra w Życie" pozwala na przetworzenie dowolnej operacji, którą możemy wykonać na tradycyjnych komputerach.

Projekt pokazał, że za pomocą ogólnie dostępnych mikrokontrolerów można przeprowadzić złożone symulacje o prostych regułach. Płytki Arduino wraz ze środowiskiem Arduino IDE oraz odpowiednimi bibliotekami są jednym z bardziej przyjaznych, oraz intuicyjnych narzędzi do stworzenia funkcjonalnego prototypu urządzenia. Udostępnione schematy płytek PCB przez Arduino pozwalają na zaprojektowanie własnego układu z użyciem wykorzystanego mikrokontrolera na danej płytce.

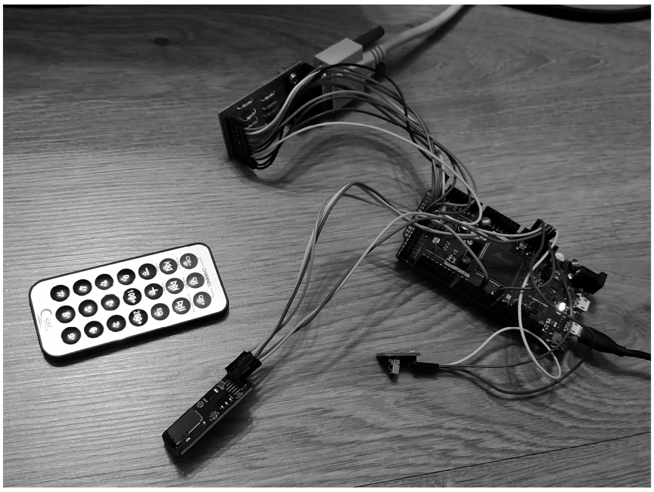
Projekt można rozwinąć o zrealizowanie przetwarzania obrazu na dodatkowej płytce FPGA, by zniwelować problem migotania obrazu. Można całkowicie przenieść symulację automatu komórkowego na FPGA — pozwalając w ten sposób na obliczanie następnego stanu komórek równolegle dla całej siatki. Rysunek 8 przedstawia układ cyfrowy pozwalający na obliczenie następnego stanu pojedynczej komórki, gdzie $a0_t$ oznacza aktualny stan komórki, wartości od $a1_t$ do $a8_t$ odpowiadają sąsiadom, $a0_{t+1}$ to stan obliczanej komórki w następnej generacji. Większość bramek logicznych tego układu odpowiada za sumowanie żywych komórek w sąsiedztwie Moore'a. Ostatnie bramki odpowiadają za sprawdzenie warunku, czy suma sąsiadów jest równa 3 lub, czy aktualna komórka jest żywa, podczas gdy liczba sąsiadów wynosi 2. Warunek ten można zapisać krócej w postaci „$S = 3$ OR $S + a0_t = 3$" [13].

Cel pracy został zrealizowany zgodnie z założeniami. Stworzone urządzenie pozwala na zapoznanie się z działaniem automatów komórkowych oraz wykonanie eksperymentów na strukturach załączonych w programie lub na karcie pamięci, jak i pozwala na zaprojektowanie własnej symulacji o dowolnych regułach. Urządzenie potrafi zasymulować dwa różne oraz niezależne od siebie automaty komórkowe co pozwala zaobserwować, jak te same struktury będą się zmieniać, kiedy zaaplikujemy im różne reguły. Dla przykładu na rysunku 14 została zaprezentowana generacja 9 dla tej samej struktury początkowej, ale o innych zasadach – po lewej „Gra w Życie", po prawej reguły „1/1" znane również pod nazwą „Narośl".

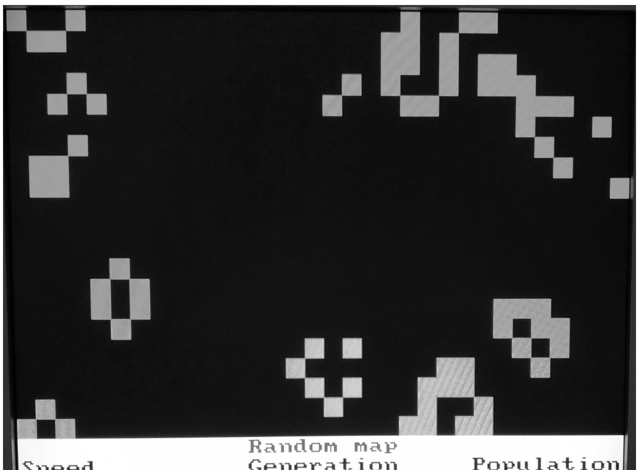


Rys. 8. Układ cyfrowy dla pojedynczej komórki automatu „Gry w Życie”

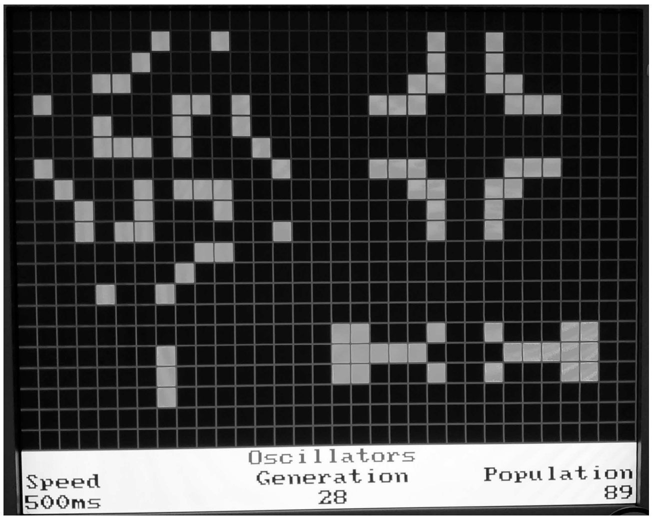
Struktura po prawej rozrasta się we wszystkich kierunkach w bardzo szybkim tempie, a w oryginalnych zasadach Conwaya rośnie bardzo powoli — wymierając już w 59 generacji oraz zostawiając cztery niezmienne struktury „Block” (klocek). Wszystkie te funkcjonalności zostały oparte o sterowanie pilotem na podczerwień oraz wyświetlanie interfejsu użytkownika poprzez złącze VGA.



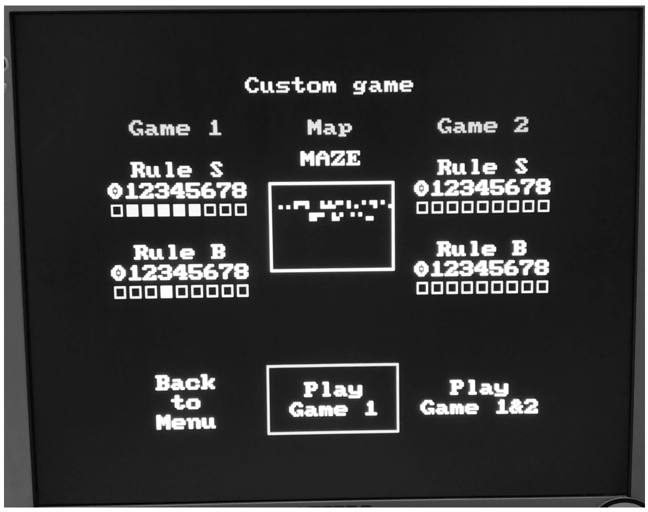
Rys. 9. Zrealizowany prototyp urządzenia



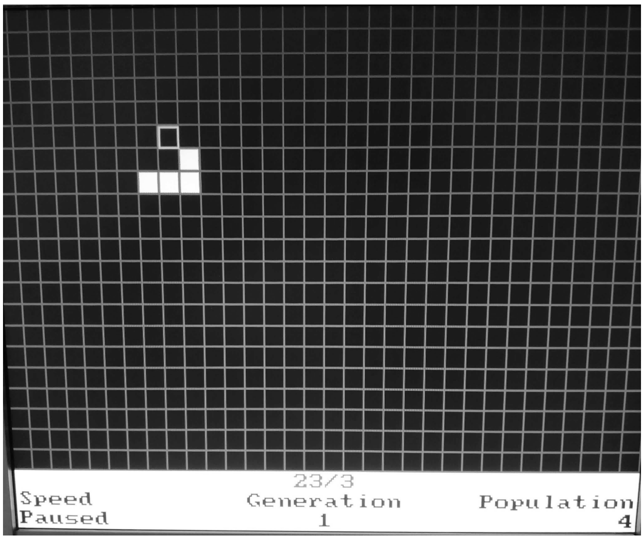
Rys. 10. Generacja 24 losowej mapy



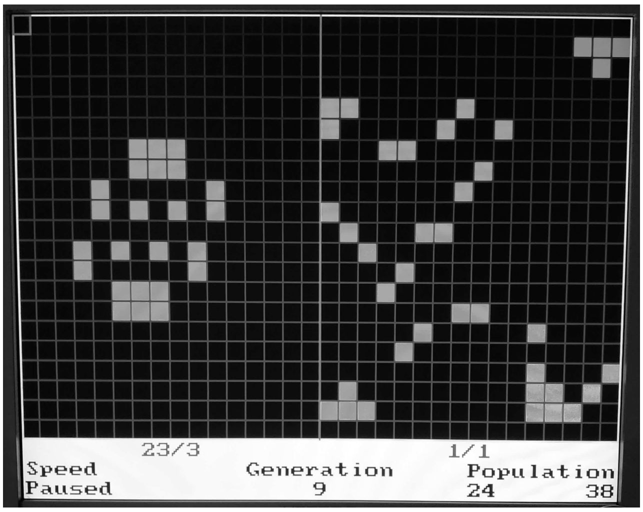
Rys. 11. Mapa z oscylatorami z podglądem siatki



Rys. 12. Kreator własnej symulacji z załadowaną mapą w podglądzie odczytaną z karty pamięci microSD



Rys. 13. Własna symulacja oraz rysowanie struktury



Rys. 14. Własna symulacja oraz rysowanie struktury

5. BIBLIOGRAFIA

1. Wolfram, Stephen (1983). "Statistical Mechanics of Cellular Automata". Reviews of Modern Physics. 55 (3): 601–644. Bibcode:1983RvMP...55..601W. doi:10.1103/RevModPhys.55.601. Archived from the original on 21 September 2013. Retrieved 28 February 2011.
2. Toffoli, Tommaso; Margolus, Norman (1987). Cellular Automata Machines: A New Environment for Modeling. MIT Press. p. 27. ISBN 9780262200608.
3. Schiff, Joel L. (2011). Cellular Automata: A Discrete View of the World. Wiley & Sons, Inc. p. 40. ISBN 9781118030639.
4. https://stimmer.github.io/DueVGA/, data dostępu 17.01.2022
5. https://en.wikipedia.org/wiki/Cellular_automaton, data dostępu 17.01.2022
6. https://en.wikipedia.org/wiki/Von_Neumann_universal_constructor, data dostępu 17.01.2022
7. Tomassini, M.; Sipper, M.; Perrenoud, M. (2000). "On the generation of high-quality random numbers by two-dimensional cellular automata". IEEE Transactions on Computers. 49 (10): 1146–1151. doi:10.1109/12.888056
8. Burraston, Dave, and Ernest Edmonds. "Cellular automata in generative electronic music and sonic art: a historical and technical review." Digital Creativity 16.3 (2005): 165-185.
9. Chowdhury, D. Roy; Basu, S.; Gupta, I. Sen; Chaudhuri, P. Pal (Czerwiec 1994). "Design of CAECC - cellular automata based error correcting code". IEEE Transactions on Computers. 43 (6): 759–764. doi:10.1109/12.286310.
10. Coombs, Stephen (15 Luty 2009), „The Geometry and Pigmentation of Seashells” (PDF), str. 3–4
11. Gardner, Martin (1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"". Scientific American. 223 (4): 120–123. doi:10.1038/scientificamerican1070-120.
12. J. Ślepecki, M. Rydzewski, P.Kisiel, P. Poczekajło, „Konsola do gier bazująca na płytce Arduino DUE”, Zeszyty Naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej, vol. 13, str. 55-61, 2018
13. https://math.stackexchange.com/questions/297433/circuit-in-conway-s-game-of-life, data dostępu 17.01.2022

SIMULATION OF JOHN CONWAY’S “GAME OF LIFE” CELLULAR AUTOMATON ON A SELECTED MICROCONTROLLER

Cellular automata are capable of simulating complex structures with a short list of simple rules. Some of them can process computer operations inside the simulation and others are used in everyday life. Each cell’s state of the automaton can be calculated based on states of their surrounding cells at the same time for the next generation. The purpose of the following thesis was to analyze cellular automata and to create a prototype device using a selected microcontroller to simulate a cellular automaton and output its visual representation on a monitor through a VGA connector – that converts digital signal from the device to analog VGA signal to the monitor. The Arduino DUE development board was used to build the prototype. The device allows the user to simulate the structures of John Conway's "Game of Life" cellular automaton and others by modifying the simulation rules. Structures can be loaded from the program menu or from a microSD memory card. The device allows the user to simulate two independent cellular automata at the same time, pause and resume simulation with the option to increase or decrease the time between generations, turning walls on or off and modify cell states. The prototype is controlled by an infrared remote control.

Keywords: cellular automaton, Game of Life, John Conway, Arduino Due.